Joint Feature Selection and Classification for Multilabel Learning

Jun Huang, Guorong Li, *Member, IEEE*, Qingming Huang, *Senior Member, IEEE*, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Multilabel learning deals with examples having multiple class labels simultaneously. It has been applied to a variety of applications, such as text categorization and image annotation. A large number of algorithms have been proposed for multilabel learning, most of which concentrate on multilabel classification problems and only a few of them are feature selection algorithms. Current multilabel classification models are mainly built on a single data representation composed of all the features which are shared by all the class labels. Since each class label might be decided by some specific features of its own, and the problems of classification and feature selection are often addressed independently, in this paper, we propose a novel method which can perform joint feature selection and classification for multilabel learning, named JFSC. Different from many existing methods, JFSC learns both shared features and label-specific features by considering pairwise label correlations, and builds the multilabel classifier on the learned low-dimensional data representations simultaneously. A comparative study with state-of-the-art approaches manifests a competitive performance of our proposed method both in classification and feature selection for multilabel learning.**

***Index Terms*—Feature selection, label correlation, label-specific features, multilabel classification, shared features.**

Manuscript received August 26, 2016; revised November 29, 2016; accepted January 23, 2017. Date of publication February 14, 2017; date of current version February 14, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61332016, Grant 61620106009, Grant U1636214, and Grant 61650202, in part by the National Basic Research Program of China (973 Program) under Grant 2015CB351800, in part by the Key Research Program of Frontier Sciences, Chinese Academy of Sciences under Grant QYZDJ-SSW-SYS013, in part by the Program for Changjiang Scholars and Innovative Research Team in University of the Ministry of Education, China under Grant IRT13059, and in part by the U.S. National Science Foundation under Grant 1652107. This paper was recommended by Associate Editor S. Ventura. *(Corresponding authors: Guorong Li; Qingming Huang.)*

J. Huang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101480, China, and also with the School of Computer Science and Technology, Anhui University of Technology, Maanshan 243032, China (e-mail: huangjun13b@mails.ucas.ac.cn).

G. Li is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101480, China (e-mail: liguorong@ucas.ac.cn).

Q. Huang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101480, China, and also with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qmhuang@ucas.ac.cn).

X. Wu is with the School of Computing and Informatics, University of Louisiana at Lafayette, Louisiana, LA 70503 USA (e-mail: xwu@louisiana.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2663838

I. Introduction

In MULTILABEL learning, each example might be associated with multiple class labels simultaneously, which may be correlated with each other. Recent works have witnessed the fast development of multilabel learning in various research areas, such as text categorization [1], [2], image annotation [3]–[5], and video annotation [6], [7]. However, with the ever-growing digital data in text, image, video, etc., the big volume and high dimensionality of data will hinder the performance of multilabel learning algorithms. Over the past years, a variety of algorithms have been proposed for multilabel learning. Most of these algorithms are designed for multilabel classification, while feature selection for multilabel learning does not attract sufficient attention. Besides, there are two limitations with existing efforts.

First, existing multilabel algorithms mainly utilize an identical data representation in the discrimination of all the class labels. In multilabel learning, each example is represented by a single instance and associated with several labels, each of which might be determined by some specific features of its own. Taking image annotation as an example, as shown in Fig. 1, each moment feature corresponds to a block region in the image. Label "sky" might be only related to the green blocks, label "ship" might be only related to the yellow blocks, and "sea water" might be only related to the blue blocks.

Second, current feature selection algorithms mainly learn a subset of features which are shared by all the class labels. As aforementioned, each class label might be decided by some specific features of its own. In addition, feature selection and multilabel classification are often addressed separately.

In order to solve the aforementioned problems, we propose to perform joint feature selection and classification (JFSC) for multilabel learning in an unified framework. We try to learn label-specific and shared features with structured sparsity regularization, and build a multilabel classifier on the learned low-dimensional data representation. In addition, we expect that the instances could become more separable in the learned low-dimensional space. Motivated by the core idea of linear discriminant analysis [8], we propose a Fisher discriminant-based regularization term to minimize the inner-class distance and maximize the intraclass distance for each label.

We summarize the contributions of this paper as follows.

1) Joint feature selection with sparsity and multilabel classification are combined into a single framework, which can select the most discriminative features for each label and learn an effective classification model. Thus, the

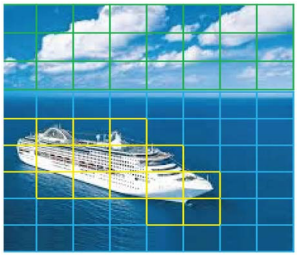

Fig. 1. Image annotation: sky, ship, and sea water.

proposed method JFSC can perform multilabel classification and feature selection simultaneously.

2) Different from existing feature selection methods, we propose to learn label-specific features and shared features by exploiting pairwise label correlation. Different from most existing multilabel classification methods, the proposed method uses label-specific data representation to discriminate each corresponding class label.
3) We propose a Fisher discriminant-based regularization term to minimize the inner-class distance and maximize the intraclass distance for each label.
4) The experiment on 16 multilabel benchmark data sets shows a competitive performance of our proposed method against the state-of-the-art multilabel learning algorithms both in classification and feature selection.

The rest of this paper is organized as follows. Section II reviews previous works on multilabel classification and feature selection. Section III presents details of the proposed method JFSC. Experimental results and analyses are shown in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

A. Multilabel Classification

In multilabel learning [9]–[13], the examples have multiple class labels simultaneously and each example is represented by one single instance. The task of multilabel learning is how to learn an effective learning function which can predict a set of possible class labels for an unseen example. Formally, let $\mathcal{X} = \mathbb{R}^m$ denote the m-dimensional input space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_l\}$ denote the label space with l class labels. Then, the task of multilabel learning is to learn a multilabel classification function $f : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ which assigns each instance $\mathbf{x} \in \mathcal{X}$ with a set of possible class labels $f(\mathbf{x}) \subseteq \mathcal{Y}$.

In the past decades, many well-established approaches have been proposed to solve multilabel classification problems in various domains. According to the popular taxonomy presented in [9], multilabel learning approaches can be divided into two categories: 1) problem transformation approaches and 2) algorithm adaption approaches.

Problem transformation approaches transform a multilabel classification problem into either one or more single-label classification problems that are solved with a single-label classification algorithm. The binary relevance (BR) approach [3] decomposes a multilabel learning problem into l independent binary (one-versus-rest) classification problems. BR is simple and effective, but it is criticized for ignoring label correlation. The label powerset (LP) approach [9] considers each unique set of labels that exists in a multilabel training data as a new label, then a multilabel classification problem is transformed into a multiclass one. LP is effective and simple and is also able to model label correlations in the training data. It will generate a larger number of new labels, and it is possible to have limited training examples for these new labels. Also, it cannot predict unseen label sets. Some approaches have been proposed to overcome these problems, such as Random k labelsets (RAkEL) [14] and multi-label classification using ensembles of pruned sets (EPS) [15].

Algorithm adaption approaches modify traditional single-label classification algorithms for multilabel classification directly. Almost all single-label classification algorithms have been revisited in order to be adapted to multilabel data, such as [2] and [16]–[18]. multi-label k-nearest neighbor (ML-kNN) [17] is derived from traditional kNN algorithm. For each new test example, its k nearest neighbors in the training data are first identified. Then, the maximum *a posteriori* rule is utilized to make prediction by reasoning with the labeling information embodied in the neighbors. RankSVM [18] adapts a maximum margin strategy to deal with multilabel data, where a set of linear classifiers are optimized to minimize the empirical ranking loss with quadratic programming and enabled to handle nonlinear cases with kernel tricks.

On the other hand, in multilabel learning, labels may be dependent on each other, not necessarily mutually exclusive. Previous works have theoretically or practically proved the effectiveness of mining such dependency relationship between labels. Over the past decades, a multitude of approaches have been proposed for multilabel classification by mining the dependency among labels. According to the way of how the dependency relationship is modeled, existing approaches can be generally grouped into three categories, i.e., *first-order*, *second-order*, and *high-order* approaches [10].

First-order approaches tackle multilabel learning without exploiting label correlation among labels, such as BR [3], multi-label learning with Label specific features (LIFT) [19], and algorithm adaption approaches ML-kNN [17] and sparse weighted instance-based multilabel (SWIM) [20]. First-order algorithms are simple and efficient, but could be less effective due to the ignorance of label correlations.

Second-order approaches tackle multilabel learning by mining *pairwise* relationship between label pairs. The most popular way to model pairwise relationship is to exploit the interaction between any pair of labels, such as calibrated label ranking [21], learning label-specific features for multilabel classification (LLSF) [22], and multi-label teaching-to-learn and learning-to-teach [23]. Another way is to incorporate the criterion of ranking loss into the objective function to be optimized when learning the classification models, e.g., RankSVM [18], back propagation for multi-label learning [2], and relative labeling-importance aware multi-label learning [24]. These algorithms exploit the dependency relationship between two labels. However, one label might be dependent on multiple class labels.

High-order approaches tackle the multilabel learning problem by mining relationships among all the class labels

or a subset of class labels, such as the transformation approaches LP [9], RAkEL [14], EPS [15], and classifier chains (CCs) [25]. CC transforms a multilabel classification problem into a chain of l binary classification subproblems, where the ith classifier f_i is trained by using the results of labels $y_1, y_2, \ldots, y_{i-1}$ as additional features. To predict subsequent labels in a given chain order, CC resorts to using outputs of the preceding classifiers $f_1, f_2, \ldots, f_{i-1}$. The performance of CC is seriously constrained by the chain order and error propagation (i.e., the incorrect predictions of preceding labels will propagate to subsequent labels), which can be alleviated by ensemble learning. In addition, it might be inappropriate that each label is dependent on all the preceding labels in a given chain order. Extended works on CC mainly search for suitable chain orders or dependent structures among class labels and reduce the complexity of inference (see [26]–[32]).

B. Multilabel Feature Selection

1) Feature Selection: Feature selection plays an important role in data mining and machine learning, as it can speed up the learning process and even boost the performance of classification. Traditionally, feature selection approaches can be grouped into three categories, i.e., filter, wrapper and embedded approaches [33]. Filter approaches select features by ranking them with correlation coefficients, such as Fisher score [8], f-statistic [34], ReliefF [35], and feature selection through message passing [36]. Wrapper approaches iteratively apply a heuristic search strategy (e.g., genetic algorithm) to determine one or more small subsets of features and evaluate their corresponding performance of classification using an off-the-shelf classifier. While embedded approaches directly incorporate feature selection as a part of the classifier training process, such as decision trees [8], neural networks [8], lasso [37], and robust feature selection (RFS) [38].

For multilabel learning, based on the taxonomy of multilabel classification algorithms, feature selection approaches can be grouped into two categories: 1) transformation-based approaches and 2) direct (adapted) approaches [39]. Problem transformation approaches transform a multilabel data instance into either one or more single-label data instances. Then, the traditional single-label feature selection approaches can be employed directly, e.g., the aforementioned filter, wrapper and embedded approaches. However, the correlation among labels in multilabel learning is usually ignored.

Direct feature selection approaches revise traditional single-label feature selection algorithms to process the multilabel data directly, including the filter, wrapper and embedded approaches, such as [40]–[50]. MReliefF and MF-statistic [40] are adapted from the traditional filter algorithms ReliefF and F-statistic. soft-constrained Laplacian score [41] is adapted from the constrained Laplacian score for semisupervised multilabel feature selection. graph-margin based multi-label feature selection algorithm [45] describes the multilabel data with a graph, and then measures the features with the large margin theory based on it. MLNB [48] is a wrapper feature selection approach for multilabel learning. It uses a genetic algorithm as the search component, and proposed a multilabel Naive Bayes classifier to select the best features. LLSF [22], subfeature uncovering with sparsity (SFUS) [49], multilabel informed feature selection (MIFS) [51], multiview multilabel [52], and convex semi-supervised multi-label feature selection [53] are embedded feature selection approaches for multilabel classification. These approaches learn linear classifiers (e.g., linear regression) with sparse regularization to conduct feature selection. Also, there are some feature extraction approaches that have been proposed for multilabel learning, such as multi-label dimensionality reduction via dependence maximization [54], the complementary decision reduct algorithm [55], and maximizes feature variance and maximizes feature-label dependence [56].

2) Label-Specific Features: Previous multilabel feature selection approaches mainly select a subset of common features shared by all the class labels. In multilabel learning, one example is associated with multiple class labels simultaneously, and each class label might be determined by some specific features of its own. Traditional feature selection approaches can be used to select a subset of *label-specific* features for each class label based on the BR transformation framework, such as lasso and F-score. However, the label correlation information among class labels is ignored.

LIFT [19] utilizes label-specific features to represent instances to predict the corresponding class label. It can be viewed as a feature mapping method, which lacks of interpretability, and does not exploit label correlation. multi-label learning approach with label-specific feature reduction based on fuzzy rough set [57] tries to select a subset of the label-specific features generated by LIFT with the fuzzy rough set approach. LLSF [22] learns label-specific features for multilabel classification by exploiting the second-order label correlation. Meanwhile, some works have been proposed to learn common and task-specific features for multitask learning, such as DirtyLasso [58], GFLasso [59], and FelMuG [60]. DirtyLasso [58] employs the $\ell_{1,\infty}$-norm and ℓ_1-norm to extract essential features shared by all the tasks and task-specific features, respectively. DirtyLasso does not exploit the correlation among different tasks. GFLasso [59] encourages highly correlated tasks to share a common set of relevant features by calculating the distance between coefficient vectors of tasks, and the ℓ_1-norm is employed to extract task-specific features. FelMuG [60] is a task sensitive feature exploration and learning approach for multitask graph classification. It aims to learn common features, task auxiliary features and task specific features. The discriminability of label (task)-specific features of these approaches are mainly modeled by the least square loss.

By surveying previous works on multilabel learning, we find that existing multilabel classification approaches mainly use an identical data representation composed of all the features of a data set to discriminate all the class labels. However, each class label might be determined by some specific features of its own. Existing multilabel feature selection approaches mainly learn a low-dimensional data representation, composed of a subset of common features shared by all the class labels. On the other hand, feature selection and multilabel classification are often addressed separately. Embedded approaches can model feature selection and classification simultaneously,

but many of them were proposed for single-label classification, e.g., lasso [37], RFS [38], and archive-based steady state micro genetic programming [61].

In this paper, we propose an unified framework for multilabel classification and feature selection. We seek to learn label-specific features and shared features for the discrimination of each class label by exploiting pairwise label correlations, and build a multilabel classifier on the learned low-dimensional data representations simultaneously.

III. Proposed Approach

A. Preliminary

For an arbitrary matrix $\mathbf{A}$, $\mathbf{a}_i$ and $\mathbf{a}^j$ are the ith row and the jth column of $\mathbf{A}$, respectively, and a_{ij} is the (i, j)th entry. Given a multilabel data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ with n examples. We denote the training data as a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n\times m}$, where m is the dimension of the data set. Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^T \in \{0, 1\}^{n\times l}$ be the label matrix, l stands for the number of class labels. $\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{il}]$ is a ground truth label vector of $\mathbf{x}_i$. If $\mathbf{x}_i$ belongs to label y_j, then $y_{ij} = 1$; otherwise $y_{ij} = 0$.

Our goal is to learn a coefficient matrix $\mathbf{W} = [\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^l] \in \mathbb{R}^{m\times l}$, which can be used for multilabel classification. Moreover, the weight of each element of $\mathbf{W}$ can guide feature selection simultaneously. To guarantee that $\mathbf{W}$ can achieve excellent performance on classification and feature selection, we expect it to possess the following properties.

1) The coefficient matrix $\mathbf{W}$ can map the data matrix $\mathbf{X}$ to the label matrix $\mathbf{Y}$ well.
2) We assume that each class label is only determined by a subset of specific features from the original feature set of a given data set. These *label-specific* features are determined by the nonzero entries of each $\mathbf{w}^i$, and have strong discriminability to the corresponding class label.
3) For each class label, we expect a large interclass distance and a small inner-class distance in the low-dimensional space with label-specific features indicated by each $\mathbf{w}^i$, $1 \le i \le l$.
4) In multilabel learning, labels may be correlated with each other. We expect that any two strongly correlated class labels can share more features with each other than two uncorrelated or weakly correlated ones.

To achieve these goals, we can generalize our problem as the following optimization formulation:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) + \alpha\mathcal{S}(\mathbf{W}) + \beta\mathcal{R}_1(\mathbf{W}) + \gamma\mathcal{R}_2(\mathbf{W}) \tag{1}$$

where $\mathcal{L}(\cdot)$ is a loss function, $\mathcal{S}(\cdot)$ is employed to model the sharing of label-specific features, $\mathcal{R}_1(\cdot)$ is the regularization term to model large interclass distances and small inner-class distances in the low-dimensional space, and $\mathcal{R}_2(\cdot)$ is the sparsity regularization term to learn label-specific features. α, β, and γ are the tradeoff parameters with non-negative values.

B. Discriminability and Sparsity of Label-Specific Features

The loss function $\mathcal{L}(\mathbf{W})$ can be implemented with various ways. Here we employ the robust loss function $\|\mathbf{x}_i\mathbf{w}^k + b_k - y_{ik}\|_2$ which is less sensitive to outliers than the squared loss $\|\mathbf{x}_i\mathbf{w}^k + b_k - y_{ik}\|_2^2$. Thus, the $\mathcal{L}(\mathbf{W})$ can be formulated as

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{2}\sum_{k=1}^{l}\left(\sum_{i=1}^{n}\left\|\mathbf{x}_i\mathbf{w}^k + b_k - y_{ik}\right\|_2\right) \tag{2}$$

where $\mathbf{b} = [b_1, b_2, \ldots, b_l]^T \in \mathbf{R}^{l\times 1}$ is the bias. For simplicity, the bias $\mathbf{b}$ can be absorbed into $\mathbf{W}$ when the constant value 1 is added as an additional dimension for each data $\mathbf{x}_i (1 \le i \le n)$. Thus $\mathcal{L}(\mathbf{W}, \mathbf{b})$ can be rewritten as $\mathcal{L}(\mathbf{W})$

$$\begin{aligned}\mathcal{L}(\mathbf{W}) &= \frac{1}{2}\sum_{k=1}^{l}\left(\sum_{i=1}^{n}\left\|\mathbf{x}_i\mathbf{w}^k - y_{ik}\right\|_2\right)\\ &= \frac{1}{2}\mathrm{Tr}\big((\mathbf{X}\mathbf{W} - \mathbf{Y})^T\mathbf{D}(\mathbf{X}\mathbf{W} - \mathbf{Y})\big)\end{aligned} \tag{3}$$

where $\mathbf{D}$ is a diagonal matrix with element d_{ii} defined as

$$d_{ii} = \frac{1}{2\|\mathbf{x}_i\mathbf{W} - \mathbf{y}_i\|_2}. \tag{4}$$

To induce sparse label-specific features, we add an ℓ_1-norm on the coefficient matrix $\mathbf{W}$

$$\mathcal{R}_2 = \|\mathbf{W}\|_1. \tag{5}$$

Thus, the nonzero entry w_{ij} indicates that the ith features is discriminative to label y_j. The larger the value of $|w_{ji}|$, the stronger the discriminability of the jth feature to y_i. We say these features are *label-specific* features to y_j.

C. Fisher Discriminant-Based Regularization

For each class label, we expect a large interclass distance between positive and negative examples, and a small inner-class distance for the positive and negative examples in the low-dimensional space with label-specific features. To reach this goal, motivated by the core idea of linear discriminant analysis [8], we propose a new Fisher discriminant-based regularization term $\mathcal{R}_1(\mathbf{W})$.

First, we define the sets of positive and negative examples in the original feature space for each label. Examples associating with the kth label are considered as positive examples, while those without this class label are considered as negative ones

$$\mathcal{P}_k = \left\{\mathbf{x}_i \middle| (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}, y_{ik} = 1\right\} \tag{6}$$

$$\mathcal{N}_k = \left\{\mathbf{x}_i \middle| (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}, y_{ik} = 0\right\} \tag{7}$$

where $\mathcal{P}_k$ and $\mathcal{N}_k$ are the sets of positive and negative examples of the kth label, respectively.

Then, the mean vectors of positive set $\mathcal{P}_k$ and negative set $\mathcal{N}_k$ in the low-dimensional space with label-specific features can be defined as $\mathbf{m}_k^+$ and $\mathbf{m}_k^-$, respectively

$$\mathbf{m}_k^+ = \frac{1}{|\mathcal{P}_k|}\sum_{\mathbf{x}_i\in\mathcal{P}_k}\mathbf{x}_i \odot \mathbf{w}^k = \bar{\mathbf{x}}_k^+ \odot \mathbf{w}^k \tag{8}$$

$$\mathbf{m}_k^- = \frac{1}{|\mathcal{N}_k|}\sum_{\mathbf{x}_i\in\mathcal{N}_k}\mathbf{x}_i \odot \mathbf{w}^k = \bar{\mathbf{x}}_k^- \odot \mathbf{w}^k \tag{9}$$

where $\mathbf{x}_i \odot \mathbf{w}^k = [x_{i1}w_{1k}, x_{i2}w_{2k}, \ldots, x_{im}w_{mk}]$ denotes an elementwise product. $\bar{\mathbf{x}}_k^+$ and $\bar{\mathbf{x}}_k^-$ are the mean vectors of positive and negative sets in the original feature space. $\bar{\mathbf{x}}_k^+ = [\bar{x}_{k1}^+, \bar{x}_{k2}^+, \ldots, \bar{x}_{km}^+]$, and $\bar{x}_{kj}^+ = (1/|\mathcal{P}_k|)\sum_{\mathbf{x}_i \in \mathcal{P}_k} x_{ij}$. $\bar{\mathbf{x}}_k^- = [\bar{x}_{k1}^-, \bar{x}_{k2}^-, \ldots, \bar{x}_{km}^-]$, and $\bar{x}_{kj}^- = (1/|\mathcal{N}_k|)\sum_{\mathbf{x}_i \in \mathcal{N}_k} x_{ij}$.

Thus, the interclass distance between positive and negative examples can be calculated by

$$S_k{}^2 = \left\|\mathbf{m}_k^+ - \mathbf{m}_k^-\right\|_2^2 = \mathbf{w}^{kT}\mathrm{diag}(\mathbf{s}_k)\mathbf{w}^k \tag{10}$$

where $\mathbf{s}_k = (\bar{\mathbf{x}}_k^+ - \bar{\mathbf{x}}_k^-) \odot (\bar{\mathbf{x}}_k^+ - \bar{\mathbf{x}}_k^-)$.

Third, the within-class distance of positive examples can be calculated by

$$S_k^{+2} = \sum_{\mathbf{x}_i \in \mathcal{P}_k} \left\|\mathbf{x}_i \odot \mathbf{w}^k - \mathbf{m}_k^+\right\|_2^2 = \mathbf{w}^{kT}\mathrm{diag}\left(\mathbf{s}_k^+\right)\mathbf{w}^k \tag{11}$$

where $\mathbf{s}_k^+ = \sum_{\mathbf{x}_i \in \mathcal{P}_k} \mathbf{s}_{ki}^+$ and $\mathbf{s}_{ki}^+ = (\mathbf{x}_i - \bar{\mathbf{x}}_k^+) \odot (\mathbf{x}_i - \bar{\mathbf{x}}_k^+)$. Similarly, the within-class distance of negative examples can be calculated by

$$S_k^{-2} = \sum_{\mathbf{x}_i \in \mathcal{N}_k} \left\|\mathbf{x}_i \odot \mathbf{w}^k - \mathbf{m}_k^-\right\|_2^2 = \mathbf{w}^{kT}\mathrm{diag}\left(\mathbf{s}_k^-\right)\mathbf{w}^k \tag{12}$$

where $\mathbf{s}_k^- = \sum_{\mathbf{x}_i \in \mathcal{N}_k} \mathbf{s}_{ki}^-$ and $\mathbf{s}_{ki}^- = (\mathbf{x}_i - \bar{\mathbf{x}}_k^-) \odot (\mathbf{x}_i - \bar{\mathbf{x}}_k^-)$.

Finally, we can define the Fisher discriminant-based regularization term as

$$\mathcal{R}_1(\mathbf{W}) = \frac{1}{2}\sum_{k=1}^{l}\left(S_k^{+2} + r_k S_k^{-2} - \lambda S_k{}^2\right) \tag{13}$$

where $r_k = |\mathcal{P}_k|/|\mathcal{N}_k|$ is used to account for potential class imbalance between positive and negative examples. λ is the tradeoff parameter between inner-class distances and intraclass distances, and is set to be $|\mathcal{P}_k|$ to allow $S_k{}^2$ to be the same order of magnitude with S_k^{+2} and $r_k S_k^{-2}$ in this paper. After some mathematical operations, (13) can be rewritten as

$$\mathcal{R}_1(\mathbf{W}) = \frac{1}{2}\sum_{k=1}^{l}\mathbf{w}^{kT}\mathbf{S}_k\mathbf{w}^k \tag{14}$$

where $\mathbf{S}_k = \mathrm{diag}(\mathbf{s}_k^+ + r_k\mathbf{s}_k^- - |\mathcal{P}_k|\mathbf{s}_k)$.

D. Sharing of Label-Specific Features

Previous works indicate that exploiting label correlation can improve the performance of multilabel classifiers. Motivated by the works on multitask learning [58]–[60], which learns the sharable features between tasks or modalities by considering their relationships, we assume that any two strongly correlated class labels can share more features with each other than two uncorrelated or weakly correlated ones. In other words, if label y_i and label y_j are strongly correlated, features discriminative to y_i may also be discriminative to y_j with a higher probability. Then the corresponding coefficients $\mathbf{w}^i$ and $\mathbf{w}^j$ will be very similar, and thus the inner product between them will be large, otherwise, the inner product will be small.

Similar to [22], we exploit the pairwise label correlation to model the property of sharing of label-specific features between label pairs by using the regularization in the following equation:

$$\mathcal{S}(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} r_{ij}\mathbf{w}^{iT}\mathbf{w}^j = \mathrm{Tr}\left(\mathbf{W}\mathbf{R}\mathbf{W}^T\right) \tag{15}$$

where $\mathbf{R} \in \mathbb{R}^{l\times l}$ with each element $r_{ij} = 1 - c_{ij}$, and c_{ij} indicates the correlation between y_i and y_j. Similar to previous works, in this paper, correlation is calculated by cosine similarity between label pairs.

IV. Optimization via Accelerated Proximal Gradient

A. Optimization

Based on the definitions of the terms regarding loss function and regularization, we can rewrite the objective function in (1) as follows:

$$\mathcal{F}(\mathbf{W}) = \frac{1}{2}\mathrm{Tr}\left((\mathbf{X}\mathbf{W} - \mathbf{Y})^T\mathbf{D}(\mathbf{X}\mathbf{W} - \mathbf{Y})\right) + \frac{\alpha}{2}\mathrm{Tr}\left(\mathbf{W}\mathbf{R}\mathbf{W}^T\right) + \frac{\beta}{2}\sum_{k=1}^{l}\mathbf{w}^{kT}\mathbf{S}_k\mathbf{w}^k + \gamma\|\mathbf{W}\|_1. \tag{16}$$

The problem (16) is convex but nonsmooth due to the ℓ_1-norm. We seek to solve it by the accelerated proximal gradient method. A general accelerated proximal gradient method can be written as the following convex optimization problem:

$$\min_{\mathbf{W}\in\mathcal{H}}\{\mathcal{F}(\mathbf{W}) = f(\mathbf{W}) + g(\mathbf{W})\} \tag{17}$$

where $\mathcal{H}$ is a real Hilbert space, $f(\mathbf{W})$ is convex and smooth, and $g(\mathbf{W})$ is convex and typically nonsmooth. $f(\mathbf{W})$ is further *Lipschitz* continuous, i.e., $\|\nabla f(\mathbf{W}_1) - \nabla f(\mathbf{W}_2)\| \le L_f\|\Delta\mathbf{W}\|$, where $\Delta\mathbf{W} = \mathbf{W}_1 - \mathbf{W}_2$, and L_f is the *Lipschitz* constant. For (16), $f(\mathbf{W})$ and $g(\mathbf{W})$ can be defined as

$$f(\mathbf{W}) = \frac{1}{2}\mathrm{Tr}\left((\mathbf{X}\mathbf{W} - \mathbf{Y})^T\mathbf{D}(\mathbf{X}\mathbf{W} - \mathbf{Y})\right) + \frac{\alpha}{2}\mathrm{Tr}\left(\mathbf{W}\mathbf{R}\mathbf{W}^T\right) + \frac{\beta}{2}\sum_{k=1}^{l}\mathbf{w}^{kT}\mathbf{S}_k\mathbf{w}^k \tag{18}$$

$$g(\mathbf{W}) = \gamma\|\mathbf{W}\|_1. \tag{19}$$

Thus, we can obtain the derivative of $f(\mathbf{W})$ with respect to $\mathbf{W}$ as

$$\nabla f(\mathbf{W}) = \mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{W} - \mathbf{X}^T\mathbf{D}\mathbf{Y} + \alpha\mathbf{W}\mathbf{R} + \mathbf{A}\odot\mathbf{W} \tag{20}$$

where $\odot$ represents the Hadamard product. $\mathbf{A} = [\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^l] \in \mathbb{R}^{m\times l}$, each element $\mathbf{a}^k = \beta(\mathbf{s}_k^+ + r_k\mathbf{s}_k^- - |\mathcal{P}_k|\mathbf{s}_k)$. Then, we have

$$\begin{aligned}
&\|\nabla f(\mathbf{W}_1) - \nabla f(\mathbf{W}_2)\|_F^2 \\
&\quad= \left\|\mathbf{X}^T\mathbf{D}\mathbf{X}\Delta\mathbf{W} + \alpha\Delta\mathbf{W}\mathbf{R} + \mathbf{A}\odot\Delta\mathbf{W}\right\|_F^2 \\
&\quad\le 3\left\|\mathbf{X}^T\mathbf{D}\mathbf{X}\Delta\mathbf{W}\right\|_F^2 + 3\|\alpha\Delta\mathbf{W}\mathbf{R}\|_F^2 + 3\|\mathbf{A}\odot\Delta\mathbf{W}\|_F^2 \\
&\quad\le 3\left\|\mathbf{X}^T\mathbf{D}\mathbf{X}\right\|_2^2\|\Delta\mathbf{W}\|_F^2 + 3\|\alpha\mathbf{R}\|_2^2\|\Delta\mathbf{W}\|_F^2 \\
&\qquad+ 3\max\left\{a_{ij}^2\right\}_{i,j}\|\Delta\mathbf{W}\|_F^2 \\
&\quad= 3\left(\left\|\mathbf{X}^T\mathbf{D}\mathbf{X}\right\|_2^2 + \|\alpha\mathbf{R}\|_2^2 + \max\left\{a_{ij}^2\right\}_{i,j}\right)\|\Delta\mathbf{W}\|_F^2.
\end{aligned}$$

Algorithm 1: JFSC for Multilabel Learning

Input: Training data matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, label matrix $\mathbf{Y} \in \mathbb{R}^{n\times l}$, and weighting parameters $\alpha, \beta, \gamma, \eta$;
Output: Model coefficient matrix $\mathbf{W}^* \in \mathbb{R}^{m\times l}$.

Initialization:
$b_0, b_1 \leftarrow 1$; $t \leftarrow 1$; $\mathbf{W}_0, \mathbf{W}_1 \leftarrow (\mathbf{X}^T\mathbf{X} + \eta\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$;
compute the matrix $\mathbf{A}$ in Eq. (20);
compute $\mathbf{R}$ by cosine similarity on $\mathbf{Y}$;
repeat
  compute the diagonal matrix $\mathbf{D}$ according to Eq. (4);
  compute L_f according to Eq. (21);
  $\mathbf{W}^{(t)} \leftarrow \mathbf{W}_t + \frac{b_{t-1}-1}{b_t}(\mathbf{W}_t - \mathbf{W}_{t-1})$;
  $\mathbf{G}^{(t)} \leftarrow \mathbf{W}^{(t)} - \frac{1}{L_f}\nabla f(\mathbf{W}^{(t)})$;
  $\mathbf{W}_t \leftarrow \mathbf{prox}_{(\gamma/L_f)}(\mathbf{G}^{(t)})$;
  $b_t \leftarrow \frac{1+\sqrt{4b_t^2+1}}{2}$;
  $t \leftarrow t+1$;
until *stop criterion reached*;
$\mathbf{W}^* \leftarrow \mathbf{W}_t$;

Here $\|\mathbf{A} \odot \Delta\mathbf{W}\|_F^2 = \sum_{i,j} a_{ij}^2 \Delta w_{ij}^2 \le \sum_{i,j} \max\{a_{ij}^2\}_{i,j} \Delta w_{ij}^2 = \max\{a_{ij}^2\}_{i,j} \sum_{i,j} \Delta w_{ij}^2 = \max\{a_{ij}^2\}_{i,j} \|\Delta\mathbf{W}\|_F^2$. Thus, $\|\mathbf{A} \odot \Delta\mathbf{W}\|_F^2 \le \max\{a_{ij}^2\}_{i,j} \|\Delta\mathbf{W}\|_F^2$.

Therefore, the *Lipschitz* constant can be calculated by

$$L_f = \sqrt{3\left(\left\|\mathbf{X}^T\mathbf{D}\mathbf{X}\right\|_2^2 + \|\alpha\mathbf{R}\|_2^2 + \max\left\{a_{ij}^2\right\}_{i,j}\right)}. \quad (21)$$

In the accelerated proximal gradient method, the accelerated proximal gradient iterates as follows:

$$\mathbf{W}^{(t)} = \mathbf{W}_t + \frac{b_{t-1}-1}{b_t}(\mathbf{W}_t - \mathbf{W}_{t-1}) \quad (22)$$

$$\mathbf{W}_t = \mathbf{prox}_\epsilon\left(\mathbf{W}^{(t)} - \frac{1}{L_f}\nabla f\left(\mathbf{W}^{(t)}\right)\right). \quad (23)$$

In [62], the work has shown that setting $\mathbf{W}^{(t)} = \mathbf{W}_t + (b_{t-1} - 1/b_t)(\mathbf{W}_t - \mathbf{W}_{t-1})$ for a sequence b_t satisfying $b_t^2 - b_t \le b_{t-1}^2$ can improve the convergence rate to $O(1/t^2)$, where $\mathbf{W}_t$ is the result of $\mathbf{W}$ at the tth iteration. ϵ is the step size, and set to be γ/L_f in this paper. The proximal operator associated with the ℓ_1-norm is the soft-thresholding operator

$$\mathbf{prox}_\epsilon(w) = (|w| - \epsilon)_+ \text{sign}(w). \quad (24)$$

Consequently, all the optimization steps of our proposed method can be summarized in Algorithm 1. After learning $\mathbf{W}$, we can obtain the prediction for a test data $\mathbf{X}_t$ by $\text{sign}(\mathbf{P}_t - \tau)$ with the given threshold τ, where $\mathbf{P}_t = \mathbf{X}_t\mathbf{W}$ and τ is set to be 0.5 in the experiments.

B. Complexity Analysis

For simplicity, we mainly analyze the complexity of the optimization parts listed in Algorithm 1. Note that $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{Y} \in \mathbb{R}^{n\times l}$, $\mathbf{R} \in \mathbb{R}^{l\times l}$, $\mathbf{A} \in \mathbb{R}^{m\times l}$, and $\mathbf{D} \in \mathbb{R}^{n\times n}$, where n is the number of instances, m is the dimensionality, and l is the number of labels.

TABLE I
EXPERIMENT DATA SETS

Data set	# Instances	# Features	# Labels	Card	Domain
cal500	502	68	174	26.044	music
genbase	645	1186	27	1.252	biology
medical	978	1449	45	1.245	text
language log	1459	1004	75	1.180	text
arts	5000	462	26	1.636	text(web)
education	5000	550	33	1.461	text(web)
recreation	5000	606	22	1.423	text(web)
science	5000	743	40	1.451	text(web)
corel5k	5000	499	374	3.522	image
rcv1(subset1)	6000	944	101	2.880	text
rcv1(subset2)	6000	944	101	2.880	text
bibtex	7395	1836	159	2.402	text
delicious	16015	500	983	19.020	text
bookmark	87856	2150	208	2.028	text
imdb	120919	1001	28	2.000	text
nuswide	269468	500	81	1.869	image

For the initialization of $\mathbf{W}_1$, the calculation consists of some operations of matrix multiplications and inversion. This leads to a complexity of $\mathcal{O}(nm^2 + m^3 + nml + m^2l)$. To calculate matrix $\mathbf{A}$, we should calculate the mean vectors, inner-class and intraclass distances for each label, and it needs $\mathcal{O}(nml)$. The correlation among labels is calculated by cosine similarity, and it needs $\mathcal{O}(nl^2)$. The most time-consuming components are steps 5, 6, and 8 in the iterations. It leads to a complexity of $\mathcal{O}(nml)$ for calculating matrix $\mathbf{D}$ in step 5. Calculating L_f in step 6 refers to matrix multiplications and singular value decomposition. Although $\mathbf{D}$ is an $n \times n$ matrix, it is diagonal and only needs a memory of $\mathcal{O}(n)$. Thus, calculating $\mathbf{X}^T\mathbf{D}\mathbf{X}$ only needs $\mathcal{O}(nm + nm^2)$, and the complexity in step 6 is $\mathcal{O}(nm^2 + nm + ml + m^3 + l^3)$. In step 8, it needs to calculate the gradient of $f(\mathbf{W})$, which leads to a complexity of $\mathcal{O}(nm^2 + m^2l + ml^2 + nml)$.

V. EXPERIMENTS

A. Experimental Configuration

We compare our proposed method JFSC with the following state-of-the-art multilabel classification and feature selection methods. The search range and configuration for the parameters of each comparing algorithm are suggested by their original published paper.

1) *JFSC (Proposed Method):* Parameters α, β, and γ are searched in $\{4^{-5}, 4^{-4}, \ldots, 4^5\}$, and η is searched in $\{0.1, 1, 10\}$. The threshold $\tau = 0.5$, and the *Lipschitz* constant L_f is calculated according to (21).
2) *BR [3]:* It decomposes the multilabel classification problem into l independent binary (one-versus-rest) classification subproblems.
3) *Ensemble CCs (ECC) [25]:* It is an ensemble version of CC, where the ensemble size m is set to be 10. The chain order for each CC is generated randomly.
4) *MLkNN*[1] *[17]:* A lazy learning approach to multilabel learning. The parameter k is searched in $\{3, 5, \ldots, 21\}$.
5) *LLSF*[2] *[22]:* Parameters α and β are searched in $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$, ρ is searched in $\{0.1, 1, 10\}$, and $\tau = 0.5$.
6) *Lasso [37]:* Least squared loss with ℓ_1-norm regularization. It learns sparse label-specific features without

[1]source code: http://cse.seu.edu.cn/PersonalPage/zhangml/index.htm.
[2]source code: http://www.escience.cn/people/huangjun/index.html.

TABLE II
EXPERIMENTAL RESULTS OF EACH COMPARING ALGORITHM ON REGULAR-SCALE DATA SETS IN TERMS OF EACH EVALUATION METRIC. ↑ (↓) INDICATES THE LARGER (SMALLER) THE VALUE, THE BETTER THE PERFORMANCE. BEST RESULTS ARE HIGHLIGHTED IN BOLD

Comparing Algorithm	cal500	genbase	medical	language log	arts	education	recreation	science
	$Hamming\ Loss \downarrow$							
JFSC	0.189±0.002	**0.000**±0.000	**0.010**±0.001	0.019±0.001	0.054±0.001	0.039±0.001	0.054±0.002	0.033±0.001
BR	**0.137**±0.003	0.001±0.001	0.011±0.001	**0.016**±0.001	**0.053**±0.002	**0.037**±0.001	**0.052**±0.001	**0.030**±0.001
ECC	0.138±0.005	0.001±0.001	0.011±0.001	**0.016**±0.001	0.056±0.001	0.041±0.002	0.055±0.005	0.032±0.001
LLSF	0.144±0.005	0.001±0.000	**0.010**±0.001	0.018±0.001	0.057±0.002	0.042±0.001	0.055±0.001	0.035±0.001
ML*k*NN	0.140±0.001	0.002±0.001	0.015±0.002	**0.016**±0.001	0.057±0.001	0.038±0.001	0.057±0.002	0.033±0.001
RFS	0.148±0.004	0.001±0.000	0.012±0.001	0.020±0.001	0.054±0.002	0.038±0.001	0.055±0.002	0.032±0.001
Lasso	**0.137**±0.004	0.001±0.001	0.011±0.001	0.021±0.004	0.054±0.001	0.038±0.001	0.054±0.001	0.031±0.001
	$Accuracy \uparrow$							
JFSC	**0.312**±0.008	**0.995**±0.003	**0.757**±0.013	**0.186**±0.009	**0.381**±0.007	**0.391**±0.017	**0.384**±0.015	**0.365**±0.018
BR	0.190±0.008	0.987±0.010	0.742±0.025	0.105±0.013	0.282±0.017	0.265±0.011	0.295±0.005	0.278±0.012
ECC	0.194±0.011	0.990±0.007	0.756±0.025	0.128±0.012	0.325±0.022	0.361±0.013	0.376±0.028	0.362±0.010
LLSF	0.263±0.014	0.993±0.004	0.756±0.019	0.145±0.012	0.374±0.017	0.368±0.014	0.352±0.007	0.350±0.012
ML*k*NN	0.201±0.007	0.978±0.008	0.609±0.033	0.063±0.007	0.207±0.014	0.241±0.025	0.223±0.019	0.209±0.014
RFS	0.228±0.009	0.992±0.005	0.717±0.028	0.136±0.020	0.317±0.016	0.323±0.011	0.315±0.011	0.296±0.006
Lasso	0.201±0.004	0.986±0.008	0.726±0.027	0.117±0.026	0.277±0.008	0.297±0.008	0.301±0.018	0.268±0.011
	$Exact\text{-}Match \uparrow$							
JFSC	0.000±0.000	**0.988**±0.006	0.669±0.020	**0.243**±0.006	**0.274**±0.009	0.274±0.021	0.306±0.013	0.261±0.021
BR	0.000±0.000	0.971±0.022	0.643±0.035	0.215±0.002	0.227±0.019	0.221±0.009	0.257±0.006	0.233±0.012
ECC	0.000±0.000	0.982±0.015	**0.689**±0.037	0.242±0.024	0.272±0.019	**0.296**±0.015	**0.335**±0.027	**0.317**±0.011
LLSF	0.000±0.000	0.986±0.009	0.668±0.025	**0.243**±0.016	0.267±0.019	0.254±0.011	0.287±0.009	0.255±0.013
ML*k*NN	0.000±0.000	0.950±0.019	0.524±0.032	0.195±0.021	0.168±0.016	0.202±0.020	0.198±0.018	0.177±0.013
RFS	0.000±0.000	0.983±0.012	0.615±0.038	0.227±0.031	0.259±0.016	0.267±0.008	0.275±0.009	0.249±0.009
Lasso	0.000±0.000	0.974±0.014	0.631±0.029	0.210±0.012	0.222±0.005	0.240±0.007	0.259±0.016	0.220±0.012
	$F_1 \uparrow$							
JFSC	**0.468**±0.009	**0.997**±0.002	**0.786**±0.013	**0.218**±0.009	**0.420**±0.008	**0.431**±0.015	**0.412**±0.015	**0.401**±0.017
BR	0.314±0.010	0.991±0.008	0.772±0.024	0.115±0.016	0.303±0.016	0.280±0.012	0.309±0.006	0.295±0.012
ECC	0.319±0.016	0.992±0.006	0.779±0.021	0.138±0.013	0.345±0.023	0.373±0.014	0.390±0.029	0.378±0.010
LLSF	0.410±0.017	0.995±0.003	**0.786**±0.020	0.161±0.012	0.414±0.017	0.408±0.015	0.375±0.008	0.384±0.013
ML*k*NN	0.330±0.009	0.984±0.006	0.638±0.034	0.067±0.007	0.222±0.014	0.254±0.027	0.232±0.019	0.221±0.015
RFS	0.364±0.012	0.994±0.003	0.752±0.024	0.155±0.023	0.338±0.016	0.344±0.012	0.330±0.013	0.313±0.006
Lasso	0.330±0.005	0.990±0.007	0.758±0.027	0.136±0.034	0.297±0.009	0.317±0.009	0.317±0.019	0.285±0.011
	$Macro\ F_1 \uparrow$							
JFSC	**0.123**±0.005	0.766±0.049	**0.366**±0.034	**0.073**±0.006	**0.243**±0.009	0.173±0.014	**0.305**±0.011	**0.213**±0.014
BR	0.040±0.002	0.730±0.030	0.350±0.018	0.050±0.002	0.186±0.006	0.157±0.026	0.248±0.016	0.169±0.007
ECC	0.039±0.002	0.750±0.033	**0.366**±0.015	0.050±0.005	0.163±0.008	0.160±0.015	0.262±0.012	0.183±0.006
LLSF	0.068±0.007	**0.769**±0.057	0.352±0.034	0.069±0.014	0.238±0.010	**0.186**±0.011	0.274±0.011	**0.213**±0.016
ML*k*NN	0.060±0.002	0.629±0.053	0.228±0.020	0.023±0.003	0.142±0.008	0.139±0.016	0.193±0.011	0.126±0.008
RFS	0.111±0.009	0.756±0.046	0.345±0.025	0.064±0.007	0.175±0.011	0.131±0.008	0.224±0.013	0.153±0.008
Lasso	0.056±0.003	0.721±0.019	0.322±0.018	0.051±0.017	0.170±0.014	0.122±0.006	0.228±0.004	0.146±0.012
	$Micro\ F_1 \uparrow$							
JFSC	**0.472**±0.010	**0.995**±0.002	**0.818**±0.013	**0.303**±0.013	**0.447**±0.006	**0.475**±0.014	**0.458**±0.014	**0.446**±0.014
BR	0.309±0.010	0.987±0.011	0.802±0.016	0.196±0.027	0.367±0.017	0.374±0.016	0.384±0.007	0.374±0.012
ECC	0.313±0.016	0.992±0.005	0.798±0.018	0.217±0.030	0.374±0.011	0.430±0.008	0.425±0.013	0.416±0.007
LLSF	0.409±0.018	0.994±0.004	0.817±0.016	0.233±0.014	0.445±0.017	0.459±0.013	0.430±0.010	0.436±0.014
ML*k*NN	0.327±0.009	0.978±0.009	0.695±0.041	0.117±0.015	0.280±0.009	0.345±0.031	0.300±0.021	0.297±0.014
RFS	0.367±0.012	0.993±0.005	0.779±0.027	0.215±0.034	0.380±0.020	0.415±0.017	0.390±0.015	0.377±0.008
Lasso	0.328±0.006	0.989±0.006	0.798±0.020	0.193±0.029	0.356±0.008	0.399±0.009	0.386±0.018	0.361±0.011

considering label correlations. The regularizer parameter is searched in $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$.

7) *RFS*[3] *[38]:* Efficient and RFS via joint $\ell_{2,1}$-norms minimization. The regularization parameter γ is tuned in $\{10^{-5}, 2^{-4}, \ldots, 10^{1}\}$.
8) *MIFS [51]:* The regularization parameters α, β and γ are tuned in $\{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.2, 0.4, 0.6, 0.8, 1, 10\}$.
9) *SFUS*[4] *[49]:* SFUS incorporates joint sparse feature selection with multilabel learning to uncover shared feature subspace. The regularization parameters α and β are tuned in $\{10^{-3}, 2^{-2}, \ldots, 10^{3}\}$.
10) *F-Score [8]:* Fisher score for feature selection.

[3]source code: http://www.escience.cn/system/file?fileId=67410.
[4]source code: http://www.escience.cn/system/file?fileId=67613.

LIBSVM [63] is utilized as the base binary learner for each binary classifier of BR and ECC, where the kernel function is set as linear kernel, and the parameter C is tuned in $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$. The experiments are conducted on 16 multilabel benchmark data sets, the details of which are summarized in Table I. "Card" indicates the average number of labels per example of a data set.

B. Evaluation Metrics

Given a testing data set $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n_t}$, where $\mathbf{y}_i \in \{0, 1\}^l$ is the ground truth labels of the ith test example, and $\hat{\mathbf{y}}_i = h(\mathbf{x}_i)$ is its predicted labels.

1) Example-Based Evaluation Metrics: We use two types of evaluation metrics, i.e., example-based and label-based [10]. They can evaluate the performance of multilabel learning algorithms from various aspects.

TABLE III
EXPERIMENTAL RESULTS OF EACH COMPARING ALGORITHM ON RELATIVE LARGE-SCALE DATA SETS IN TERMS OF EACH EVALUATION METRIC. ↑ (↓) INDICATES THE LARGER (SMALLER) THE VALUE, THE BETTER THE PERFORMANCE. BEST RESULTS ARE HIGHLIGHTED IN BOLD

Comparing Algorithm	corel5k	rcv1(subset1)	rcv1(subset2)	bibtex	delicious	bookmark	imdb	nuswide
	$Hamming\ Loss \downarrow$							
JFSC	**0.009**±0.000	0.027±0.000	**0.023**±0.000	0.013±0.001	0.020±0.000	**0.009**±0.000	0.081±0.000	0.027±0.000
BR	**0.009**±0.000	**0.026**±0.000	**0.023**±0.000	**0.012**±0.000	**0.018**±0.000	0.017±0.000	0.073±0.000	**0.022**±0.000
ECC	0.012±0.000	0.029±0.000	0.025±0.001	0.013±0.000	0.031±0.000	0.016±0.000	0.085±0.000	**0.022**±0.000
LLSF	0.012±0.000	0.029±0.000	0.025±0.001	0.013±0.000	0.019±0.000	**0.009**±0.000	0.085±0.000	0.030±0.001
ML*k*NN	**0.009**±0.000	**0.026**±0.000	0.024±0.000	**0.012**±0.000	**0.018**±0.000	**0.009**±0.000	**0.071**±0.000	**0.022**±0.000
RFS	**0.009**±0.000	**0.026**±0.001	**0.023**±0.000	**0.012**±0.000	**0.018**±0.000	**0.009**±0.000	0.072±0.000	**0.022**±0.000
Lasso	**0.009**±0.000	**0.026**±0.001	**0.023**±0.001	**0.012**±0.000	**0.018**±0.000	**0.009**±0.000	0.072±0.000	**0.022**±0.000
	$Accuracy \uparrow$							
JFSC	**0.144**±0.005	**0.355**±0.004	0.390±0.008	0.355±0.014	**0.221**±0.002	0.256±0.003	**0.250**±0.003	**0.189**±0.002
BR	0.039±0.006	0.253±0.006	0.317±0.007	0.346±0.012	0.116±0.003	0.269±0.001	0.068±0.001	0.079±0.001
ECC	0.118±0.009	0.339±0.008	**0.411**±0.013	0.332±0.006	0.154±0.001	**0.271**±0.001	0.083±0.001	0.078±0.017
LLSF	**0.144**±0.007	0.351±0.005	0.356±0.007	**0.360**±0.011	0.201±0.002	0.257±0.003	0.243±0.002	0.172±0.001
ML*k*NN	0.041±0.005	0.274±0.008	0.281±0.007	0.283±0.003	0.129±0.005	0.236±0.004	0.006±0.000	0.064±0.001
RFS	0.064±0.003	0.268±0.006	0.322±0.007	0.308±0.004	0.132±0.002	0.201±0.002	0.071±0.001	0.044±0.001
Lasso	0.055±0.006	0.252±0.011	0.306±0.013	0.303±0.007	0.123±0.003	0.191±0.002	0.035±0.001	0.071±0.001
	$Exact\text{-}Match \uparrow$							
JFSC	0.205±0.008	0.049±0.004	0.182±0.011	0.151±0.016	0.000±0.000	0.209±0.003	0.071±0.002	0.168±0.002
BR	0.053±0.008	0.083±0.004	0.208±0.009	0.187±0.013	**0.002**±0.000	0.200±0.001	0.030±0.001	0.227±0.001
ECC	0.167±0.013	**0.222**±0.007	**0.328**±0.004	**0.189**±0.000	0.001±0.002	0.199±0.001	0.060±0.000	**0.236**±0.003
LLSF	**0.208**±0.011	0.051±0.006	0.173±0.014	0.178±0.015	0.000±0.000	0.208±0.004	**0.077**±0.002	0.169±0.005
ML*k*NN	0.056±0.006	0.090±0.011	0.156±0.010	0.161±0.004	0.001±0.000	**0.211**±0.003	0.004±0.000	0.213±0.000
RFS	0.091±0.004	0.110±0.003	0.215±0.006	0.179±0.005	**0.002**±0.001	0.188±0.002	0.036±0.000	0.229±0.001
Lasso	0.078±0.007	0.079±0.007	0.189±0.014	0.171±0.012	0.001±0.000	0.177±0.001	0.016±0.001	0.230±0.003
	$F_1 \uparrow$							
JFSC	0.013±0.003	**0.460**±0.004	**0.466**±0.007	**0.434**±0.014	**0.342**±0.003	0.275±0.003	**0.325**±0.003	**0.238**±0.002
BR	0.006±0.001	0.320±0.007	0.361±0.006	0.404±0.011	0.181±0.004	0.301±0.002	0.083±0.001	0.100±0.001
ECC	**0.015**±0.003	0.390±0.010	0.444±0.017	0.387±0.008	0.243±0.001	**0.304**±0.002	0.092±0.001	0.094±0.020
LLSF	0.008±0.001	0.456±0.006	0.426±0.006	0.426±0.011	0.306±0.003	0.277±0.003	0.311±0.002	0.216±0.001
ML*k*NN	0.005±0.001	0.346±0.007	0.331±0.008	0.332±0.004	0.201±0.007	0.246±0.004	0.007±0.000	0.083±0.001
RFS	0.006±0.000	0.332±0.008	0.366±0.008	0.359±0.004	0.205±0.003	0.206±0.002	0.085±0.001	0.056±0.001
Lasso	0.005±0.003	0.320±0.012	0.353±0.014	0.355±0.007	0.191±0.004	0.196±0.002	0.043±0.001	0.091±0.002
	$Macro\ F_1 \uparrow$							
JFSC	0.038±0.002	**0.260**±0.009	**0.227**±0.008	**0.357**±0.005	0.104±0.002	0.138±0.002	**0.082**±0.001	**0.045**±0.000
BR	0.019±0.003	0.189±0.009	0.172±0.008	0.303±0.009	0.066±0.002	0.179±0.003	0.059±0.001	0.018±0.000
ECC	0.025±0.003	0.202±0.011	0.205±0.009	0.296±0.005	**0.113**±0.002	**0.197**±0.002	0.035±0.000	0.019±0.003
LLSF	**0.039**±0.002	0.251±0.004	0.205±0.008	0.328±0.003	0.093±0.003	0.142±0.002	0.078±0.001	0.043±0.001
ML*k*NN	0.021±0.003	0.184±0.005	0.142±0.005	0.206±0.005	0.064±0.002	0.135±0.002	0.011±0.002	0.022±0.000
RFS	0.019±0.001	0.132±0.007	0.117±0.007	0.213±0.007	0.056±0.001	0.066±0.001	0.028±0.001	0.010±0.000
Lasso	0.017±0.001	0.131±0.006	0.117±0.006	0.222±0.009	0.052±0.001	0.065±0.001	0.020±0.001	0.015±0.000
	$Micro\ F_1 \uparrow$							
JFSC	0.243±0.010	**0.498**±0.004	**0.480**±0.010	0.476±0.008	**0.369**±0.003	**0.312**±0.002	**0.343**±0.003	**0.363**±0.003
BR	0.076±0.012	0.366±0.006	0.385±0.007	0.461±0.014	0.196±0.004	0.245±0.003	0.115±0.001	0.216±0.001
ECC	0.180±0.013	0.395±0.008	0.418±0.016	0.431±0.002	0.240±0.002	0.250±0.003	0.091±0.001	0.193±0.029
LLSF	**0.244**±0.012	0.495±0.005	0.443±0.003	**0.490**±0.008	0.343±0.003	0.308±0.003	0.330±0.002	0.316±0.005
ML*k*NN	0.081±0.008	0.386±0.007	0.353±0.009	0.381±0.006	0.220±0.007	0.272±0.004	0.010±0.000	0.179±0.000
RFS	0.122±0.004	0.368±0.010	0.384±0.008	0.404±0.006	0.224±0.002	0.224±0.002	0.110±0.002	0.132±0.002
Lasso	0.106±0.008	0.365±0.014	0.383±0.013	0.406±0.007	0.212±0.003	0.217±0.003	0.063±0.001	0.202±0.003

1) *Hamming loss* evaluates how many times an example-label pair is misclassified. $[\![x]\!]$ is an indication function, it returns 1 if x holds and 0, otherwise. The smaller the value of Hamming loss, the better performance of the classifier

$$\text{Hamming loss} = \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{1}{l}\sum_{j=1}^{l}[\![y_{ij} \neq \hat{y}_{ij}]\!].$$

2) *Accuracy* evaluates Jaccard similarity between the predicted labels and the ground truth labels

$$\text{Accuracy} = \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{|\mathbf{y}_i \wedge \hat{\mathbf{y}}_i|}{|\mathbf{y}_i \vee \hat{\mathbf{y}}_i|}.$$

3) *Exact-Match* evaluates how many times the prediction and the ground truth are exactly matched

$$\text{Exact-Match} = \frac{1}{n_t}\sum_{i=1}^{n_t}[\![\mathbf{y}_i = \hat{\mathbf{y}}_i]\!].$$

4) F_1 is the harmonic mean of recall and precision. p_i and r_i are the precision and recall for the ith example

$$F_1 = \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{2p_i r_i}{p_i + r_i}.$$

2) Label-Based Evaluation Metrics: The label-based evaluation metrics are defined as

$$\text{Macro } B(h) = \frac{1}{l}\sum_{q=1}^{l} B\big(\text{TP}_q, \text{FP}_q, \text{TN}_q, \text{FN}_q\big)$$

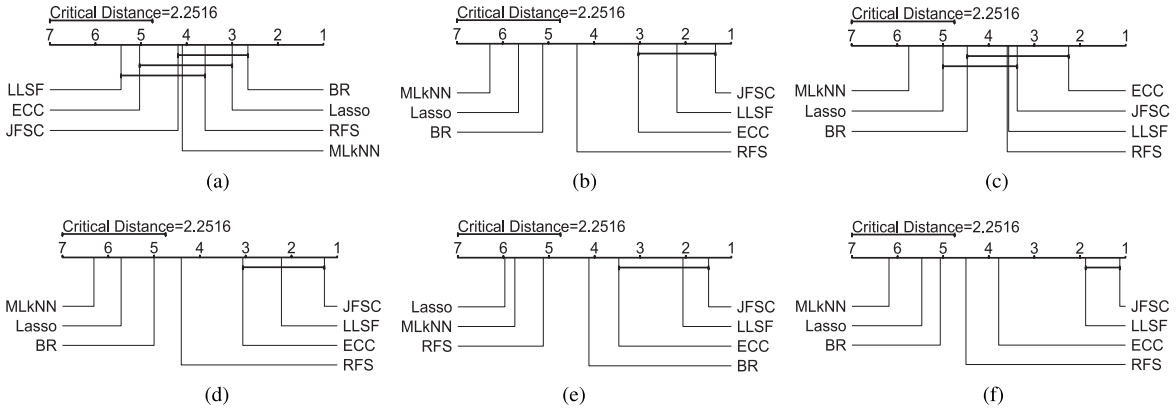

Fig. 2. Comparison of JFSC (control algorithm) against other comparing algorithms with the Nemenyi test. Groups of classifiers that are not significantly different from JFSC (at $\alpha = 0.05$) are connected. (a) Hamming loss. (b) Accuracy. (c) Exact-match. (d) F_1. (e) Macro F_1. (f) Micro F_1.

$$\text{Micro } B(h) = B\left(\sum_{q=1}^{l} \text{TP}_q, \sum_{q=1}^{l} \text{FP}_q, \sum_{q=1}^{l} \text{TN}_q, \sum_{q=1}^{l} \text{FN}_q\right)$$

where TP_q, FP_q, TN_q, and FN_q represent the number of true positive, false positive, true negative, and false negative test examples, respectively, with respect to label y_q, and $B(\text{TP}_q, \text{FP}_q, \text{TN}_q, \text{FN}_q)$ indicates some specific binary classification metrics (e.g., F_1).

TABLE IV
SUMMARY OF THE FRIEDMAN STATISTICS $F_F(k = 7, N = 16)$ AND THE CRITICAL VALUE IN TERMS OF EACH EVALUATION METRIC (k: # COMPARING ALGORITHMS; N: # DATA SETS)

Metric	F_F	Critical Value ($\alpha = 0.05$)
$Hamming\ Loss$	4.2172	2.2011
$Accuracy$	42.0095	
$Exact\text{-}Match$	6.0968	
F_1	44.2397	
$Macro\ F_1$	29.2378	
$Micro\ F_1$	46.3174	

C. Application to Multilabel Classification

In this section, we compare JFSC with BR [3], ECC [25], MLkNN [17], LLSF [22], Lasso [37], and RFS [38]. JFSC, LLSF, Lasso, and RFS can be utilized for multilabel classification and feature selection. For multilabel classification, the embedded linear classifiers (i.e., linear regression) of these approaches are employed in classification.

For each comparing algorithm, fivefold cross-validation is performed on the training data of each data set. Tables II and III report the average results (mean±std) of each comparing algorithm over 16 data sets in terms of each evaluation metric. To analyze the relative performance among the comparing algorithms systematically, Friedman test [64] is employed to conduct performance analysis. Table IV summarizes the Friedman statistics F_F and the corresponding critical value in terms of each evaluation metric. As shown in Table IV, at significance level $\alpha = 0.05$, the null hypothesis that all the comparing algorithms perform equivalently is clearly rejected in terms of each evaluation metric. Consequently, we can proceed with a post-hoc test [64] to analyze the relative performance among the comparing algorithms.

The Nemenyi test [64] is employed to test whether our proposed method achieves a competitive performance against the comparing algorithms, where JFSC is considered as the control algorithm. The performance between two classifiers will be significantly different if their average ranks differ by at least one critical difference $\text{CD} = q_\alpha \sqrt{(k(k+1)/6N)}$. For Nemenyi test, $q_\alpha = 2.948$ at significance level $\alpha = 0.05$, and thus $\text{CD} = 2.2516$ ($k = 7, N = 16$). Fig. 2 shows the CD diagrams on each evaluation metric. In each subfigure, any comparing algorithm whose average rank is within one CD to that of JFSC is connected. Otherwise, any algorithm not connected with JFSC is considered to have significantly different performance between them.

Based on these experimental results, the following observations can be made.

1) All the *first-order* approaches (i.e., BR, Lasso, RFS, and MLkNN) achieve better performance on *Hamming loss* [see Fig. 2(a)] than *second-order* approach (i.e., JFSC and LLSF) and *high-order* approaches (i.e., ECC), as *first-order* approaches try to optimize Hamming loss. JFSC achieves better performance than LLSF and ECC in terms of *Hamming loss*.
2) All the *second-order* approaches (i.e., JFSC and LLSF) and *high-order* approach (i.e., ECC) achieve better performance on *Exact-Match* [see Fig. 2(c)] than the *first-order* approaches (i.e., BR, Lasso, RFS, and MLkNN). As previous works suggest that optimizing *Exact-Match* need to model label correlations. JFSC achieves better performance than LLSF in terms of *Exact-Match*.
3) JFSC significantly outperforms Lasso, BR, MLkNN, and RFS in terms of other four evaluation metrics. Lasso can be regarded as a plain version of JFSC by learning label-specific features without considering label correlation and innerclass and intraclass distances for each label. The super performance of JFSC against these approaches indicates the effectiveness of learning label-specific features and exploiting label correlation.
4) Furthermore, JFSC performs worse than ECC in terms of *Exact-Match*, and obtains statistically superior performance against ECC in terms of the other five evaluation metrics.

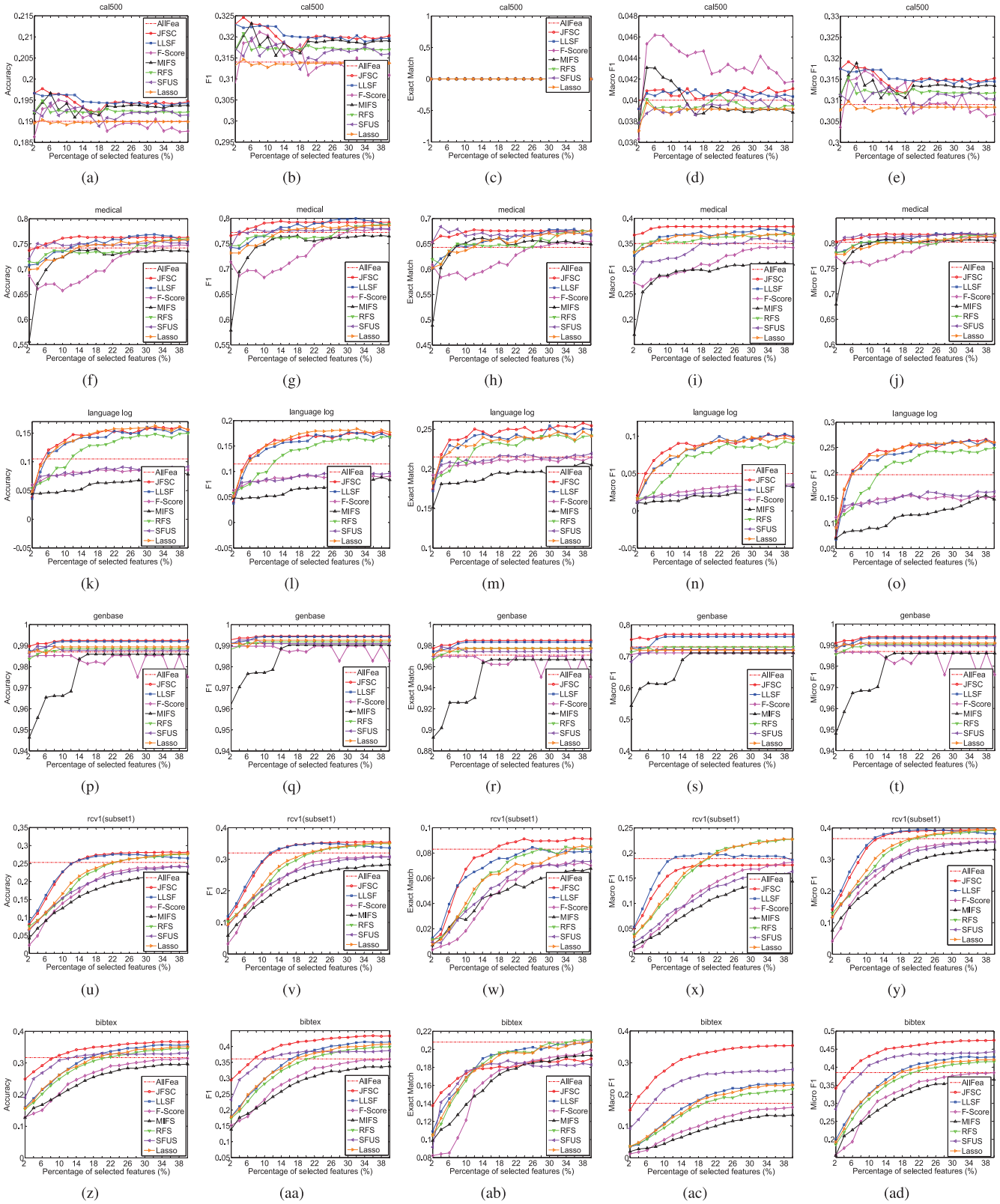

Fig. 3. Results of feature selection: BR with LIBSVM is employed as the multilabel classifier. cal500: (a) Accuracy (b) F_1, (c) Exact-match, (d) Macro F_1, and (e) Micro F_1. medical: (f) Accuracy, (g) F_1, (h) Exact-match, (i) Macro F_1, and (j) Micro F_1. language log: (k) Accuracy, (l) F_1, (m) Exact-match, (n) Macro F_1, and (o) Micro F_1. genbase: (p) Accuracy, (q) F_1, (r) Exact-match, (s) Macro F_1, and (t) Micro F_1. rcv1(subset1): (u) Accuracy, (v) F_1, (w) Exact-match, (x) Macro F_1, and (y) Micro F_1. bibtex: (z) Accuracy, (aa) F_1, (ab) Exact-match, (ac) Macro F_1, and (ad) Micro F_1.

5) JFSC achieves statistically superior performance against LLSF in terms of each evaluation metric. The better performance of JFSC against LLSF indicates the effectiveness of utilizing a robust loss function and modeling innerclass and intraclass distances.

TABLE V
RESULTS OF PAIRWISE COMPARISON APPLIED TO JFSC WITH COMPARING ALGORITHMS

Comparing Algorithm	$Accuracy$			F_1			$Exact\text{-}Match$			$Macro\ F_1$			$Micro\ F_1$			$N/2 + 1.96\sqrt{N}/2$ ($N = 126$)
	win	tie	lose	win	tie	lose	win	tie	lose	win	tie	lose	win	tie	lose	
AllFea	113	0	13	113	0	13	75	21	30	102	0	24	116	0	10	74.0005
LLSF	101	1	24	98	1	27	81	21	24	89	1	36	95	1	30	
F-Score	124	0	2	124	0	2	93	21	12	104	0	22	123	0	3	
MIFS	123	0	3	123	0	3	98	21	7	121	0	5	125	0	1	
RFS	124	0	2	123	0	3	85	21	20	113	0	13	123	0	3	
SFUS	123	0	3	123	0	3	92	22	12	126	0	0	116	0	10	
Lasso	114	0	12	109	0	17	86	21	19	109	0	17	112	0	14	
Total	822	1	59	813	1	68	610	148	124	764	1	117	810	1	71	

To summarize, JFSC achieves a competitive performance against other well-established multilabel classification approaches.

D. Application to Multilabel Feature Selection

In this section, we employ BR with LIBSVM [63] as the multilabel classifier to evaluate the performance of JFSC and other feature selection approaches on multilabel feature selection.

We compare JFSC with F-Score [8], Lasso [37], LLSF [22], RFS [38], MIFS [51], and SFUS [49]. Parameters for each comparing algorithm are searched by fivefold cross-validation on the training data according the performance of themselves on multilabel classification (i.e., the linear classifiers). Features are selected according to the absolute weight of the coefficient (or score) matrix of these approaches. Given a percentage of features to be selected, for each label, a low-dimensional data representation, composed of the features with top weights, is taken as the input data for the corresponding binary classifier of BR. LIBSVM with linear kernel is initialized as the one-versus-rest binary classifier for each label, and the parameter C is tuned in $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$.

The experiments are conducted on six data sets, i.e., cal500, genbase, medical, language log, rcv1subset1, and bibtex. For each data set, we randomly split it into training (80%) and testing (20%) parts, and the number of selected features in varied in top $\{2\%, 4\%, \ldots, 40\%\}$ of the total number of features. The average result of five times repetitions of each comparing algorithm on different data sets are reported in Fig. 3. "AllFea" means that the original data with no feature selection is used as a baseline, and its performance is equal to BRs. As the number of selected features is varied from top 2% to 40% with a step of 2%, there are 21 points totally. Table V summarizes the overall pairwise comparison results of JFSC with other compared feature selection approaches over the six data sets in terms of each evaluation metric. Each cell is composed of three numbers: from left to right, how many times that JFSC achieves a better/tied/worse performance than the compared approach at different percentages of selected features with a given evaluation metric. For example, in the third row and the second column, "113 0 13" means that JFSC wins AllFea 113 times, ties 0 time, and loses 13 times over the six data sets ($21 \times 6 = 126$ points totally) in terms of *Accuracy*. The sign test [64] is employed to test whether JFSC achieves a competitive performance against the other comparing algorithms. If the number of wins is at least $N/2+1.96\sqrt{N}/2$, the algorithm is significantly better with significance level $\alpha < 0.05$, where $N = 126$. According to the experimental results, we can make the following observations.

1) These feature selection methods generally perform better than AllFea which does not conduct feature selection. This observation indicates that feature selection contributes to improvement of multilabel classification performance.
2) JFSC and LLSF generally obtain better performance than other feature selection methods. The advantage indicates the effectiveness of learning label-specific features for each class label in multilabel learning.
3) JFSC consistently outperforms the other feature selection methods in terms of each evaluation metric. The advantage indicates the effectiveness of our proposed method in feature selection for multilabel learning.

To summarize, JFSC achieves a competitive performance against other feature selection approaches.

E. Parameter Sensitivity Analysis

To conduct parameter sensitivity analysis of JFSC, we first find a group of best configurations for the parameters by cross-validation on the training data of the rcv1(subset1) data set, and then fix the value of one parameter and vary the values of the other two parameters.

We randomly split the data set into training (80%) and testing (20%) parts five times, the average results of JFSC over the five repetitions with different values of α, β, and γ are depicted from Fig. 4(a)–(o). We can observe that the highest performance is achieved at some intermediate values of α, β, and γ. The experiments show that the performance of JFSC is sensitive to the values of the regularization parameters. However, large candidate sets for α, β, and γ can be employed in practice to obtain satisfactory performance. With a fixed setting of α, β, and γ, we evaluate the influence of η to the performance of JFSC over five data sets. The experimental results are shown from Fig. 4(p)–(t). The performance of JFSC is improved slightly and then declines with the increase of η, and the best results are obtained at $\eta = 1$ in most cases.

F. Discussion

From the experimental results shown in Fig. 4, the performance of JFSC is sensitive to the parameter configuration. An appropriate configuration of parameters could be obtained by cross-validation on the training data. For large-scale data sets, it will cost significant amount of time.

According the time complexity analysis (see Section IV-B), we find that the complexities of the initialization of $\mathbf{W}_1$

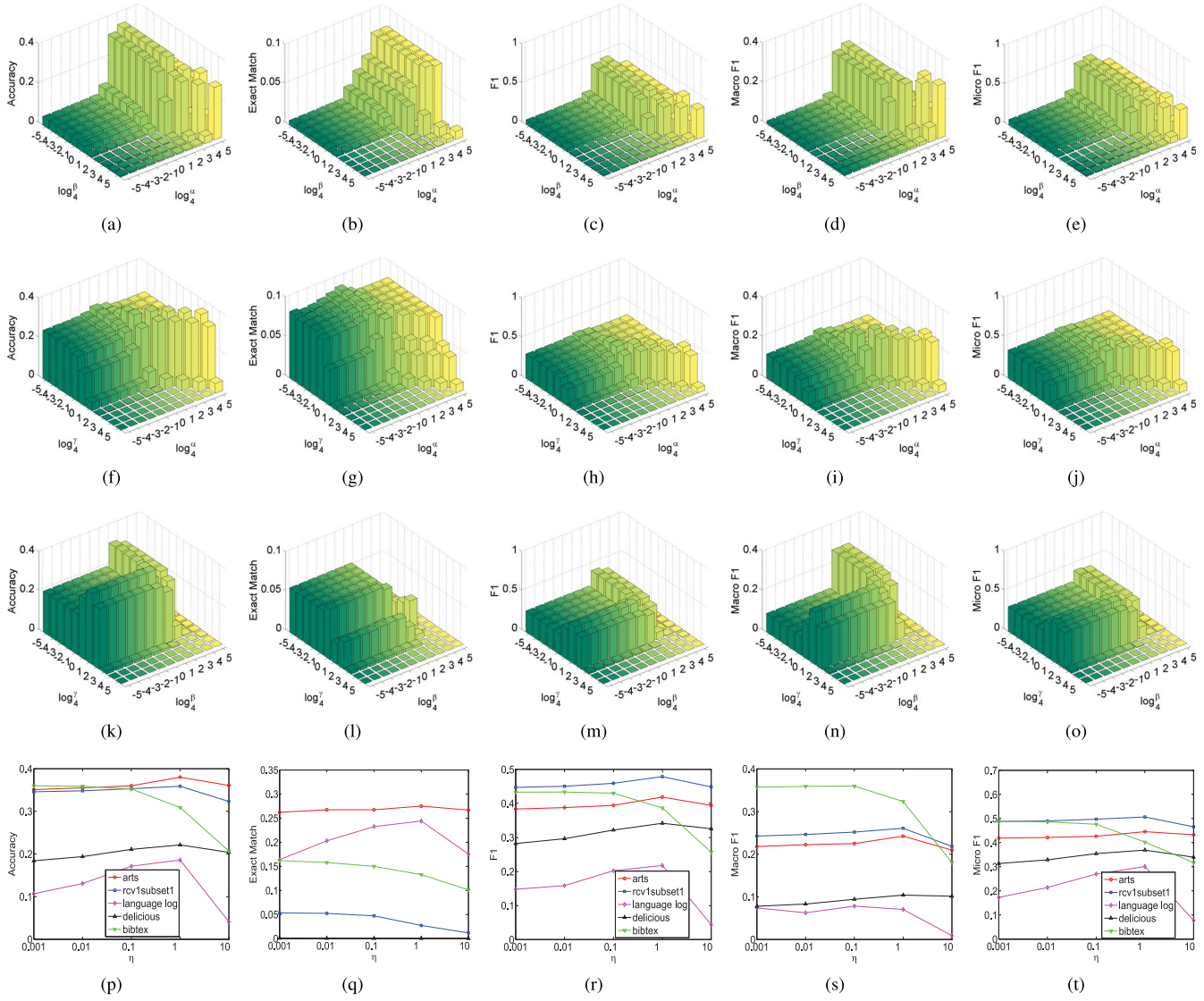

Fig. 4. Parameter sensitivity analysis of JFSC. (a) Accuracy of JFSC with fixed γ. (b) Exact-match of JFSC with fixed γ. (c) F_1 of JFSC with fixed γ. (d) Macro F_1 of JFSC with fixed γ. (e) Micro F_1 of JFSC with fixed γ. (f) Accuracy of JFSC with fixed β. (g) Exact-match of JFSC with fixed β. (h) F_1 of JFSC with fixed β. (i) Macro F_1 of JFSC with fixed β. (j) Micro F_1 of JFSC with fixed β. (k) Accuracy of JFSC with fixed α. (l) Exact-match of JFSC with fixed α. (m) F_1 of JFSC with fixed α. (n) Macro F_1 of JFSC with fixed α. (o) Micro F_1 of JFSC with fixed α. (p) Accuracy of JFSC with fixed α, β and γ. (q) Exact-match of JFSC with fixed α, β and γ. (r) F_1 of JFSC with fixed α, β and γ. (s) Macro F_1 of JFSC with fixed α, β and γ. (t) Micro F_1 of JFSC with fixed α, β and γ.

(step 1) and the calculation of the *Lipschitz* constant L_f (step 6) are cubic with respect to the number of features m and the number of labels l. This would indeed make our proposed approach less scalable to multilabel data sets with large numbers of features. Actually, if the initialization of $\mathbf{W}_1$ is solved by the gradient descent algorithm, then its complexity will decrease to quadratic. Moreover, an appropriate $L \geq L_f$ can be searched by a backtracking stepsize rule [62] and used instead of L_f. Then the complexity of step 6 will also decrease to quadratic.

VI. Conclusion

In this paper, we proposed an unified framework which can perform JFSC for multilabel learning. We proposed to learn label-specific features and shared features for the discrimination of each class label by exploiting pairwise label correlations, and then build a multilabel classifier on the low-dimensional data representations composed of these learned features. The experiments verified the usefulness of exploiting label correlation and learning label-specific data representation for multilabel learning. A comparative study with state-of-the-art approaches manifested a competitive performance of our proposed method both in classification and feature selection for multilabel learning.

References

[1] A. K. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Proc. AAAI Workshop Text Learn.*, Orlando, FL, USA, 1999.

[2] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[3] M. R. Boutell, J.-B. Luo, X.-P. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[4] Y. Xia *et al.*, "Weakly supervised multilabel clustering and its applications in computer vision," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3220–3232, Dec. 2016.

[5] R. Hong *et al.*, "Image annotation by multiple-instance learning with discriminative feature mapping and selection," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 669–680, May 2014.

[6] F. Kang, R. Jin, and R. Sukthankar, "Correlated label propagation with application to multi-label learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, 2006, pp. 1719–1726.

[7] G.-J. Qi *et al.*, "Correlative multi-label video annotation," in *Proc. ACM Multimedia*, Augsburg, Germany, 2007, pp. 17–26.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2000.

[9] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2010, pp. 667–685.

[10] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[11] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Comput. Surveys*, vol. 47, no. 3, pp. 1–38, 2015.

[12] F. Herrera, F. C. Charte, A. J. Rivera, and M. J. del Jesus, *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Cham, Switzerland: Springer, 2016.

[13] E. Gibaja and S. Ventura, "Multi-label learning: A review of the state of the art and ongoing research," *Wiley Interdisc. Rev. Data Min. Knowl. Disc.*, vol. 4, no. 6, pp. 411–444, 2014.

[14] G. Tsoumakas and I. Vlahavas, "Random *k*-labelsets: An ensemble method for multilabel classification," in *Proc. Eur. Conf. Mach. Learn.*, Warsaw, Poland, 2007, pp. 406–417.

[15] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Proc. IEEE Int. Conf. Data Min.*, Pisa, Italy, 2008, pp. 995–1000.

[16] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Proc. Data Min. Knowl. Disc.*, Freiburg im Breisgau, Germany, 2001, pp. 42–53.

[17] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[18] A. Elisseeff and W. Jason, "A kernel method for multi-labelled classification," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2001, pp. 681–687.

[19] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, Jan. 2015.

[20] H. Liu, X. Li, and S. Zhang, "Learning instance correlation functions for multilabel classification," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 499–510, Feb. 2017.

[21] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, 2008.

[22] J. Huang, G.-R. Li, Q.-M. Huang, and X.-D. Wu, "Learning label specific features for multi-label classification," in *Proc. IEEE Int. Conf. Data Min.*, Atlantic City, NJ, USA, 2015, pp. 181–190.

[23] C. Gong, D. Tao, J. Yang, and W. Liu, "Teaching-to-learn and learning-to-teach for multi-label propagation," in *Proc. AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 1610–1616.

[24] Y.-K. Li, M.-L. Zhang, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," in *Proc. IEEE Int. Conf. Data Min.*, Atlantic City, NJ, USA, 2015, pp. 251–260.

[25] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Eur. Conf. Mach. Learn.*, Bled, Slovenia, 2009, pp. 254–269.

[26] W. Bi and J. T. Kwok, "Bayes-optimal hierarchical multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 2907–2918, Nov. 2015.

[27] W. Cheng, E. Hüllermeier, and K. J. Dembczyński, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 279–286.

[28] J. H. Zaragoza, L. E. Sucar, E. F. Morales, C. Bielza, and P. Larrañaga, "Bayesian chain classifiers for multidimensional classification," in *Proc. Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 2192–2197.

[29] A. Kumar, S. Vembu, A. K. Menon, and C. Elkan, "Beam search algorithms for multilabel learning," *Mach. Learn.*, vol. 92, no. 1, pp. 65–89, 2013.

[30] J. Read, L. Martino, and D. Luengo, "Efficient Monte Carlo methods for multi-dimensional learning with classifier chains," *Pattern Recognit.*, vol. 47, no. 3, pp. 1535–1546, 2014.

[31] F. Briggs, X. Z. Fern, and R. Raich, "Context-aware MIML instance annotation: Exploiting label correlations with classifier chains," *Knowl. Inf. Syst.*, vol. 43, no. 1, pp. 53–79, 2015.

[32] J. Huang, G.-R. Li, Q.-M. Huang, and X.-D. Wu, "Learning label-specific features and class-dependent labels for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3309–3323, Dec. 2016.

[33] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[34] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. New York, NY, USA: Springer, 1998.

[35] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *Proc. Eur. Conf. Mach. Learn.*, Catania, Italy, 1994, pp. 171–182.

[36] P. P. Kundu and S. Mitra, "Feature selection through message passing," *IEEE Trans. Cybern.*, to be published.

[37] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[38] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2010, pp. 1813–1821.

[39] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artif. Intell. Rev.*, Sep. 2016, to be published, doi: 10.1007/s10462-016-9516-4.

[40] D. Kong, C. Ding, H. Huang, and H. Zhao, "Multi-label reliefF and F-statistic feature selections for image annotation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 2352–2359.

[41] A. Alalga, K. Benabdeslem, and N. Taleb, "Soft-constrained Laplacian score for semi-supervised multi-label feature selection," *Knowl. Inf. Syst.*, vol. 47, no. 1, pp. 75–98, 2016.

[42] X. Kong and P. S. Yu, "gMLC: A multi-label feature selection framework for graph classification," *Knowl. Inf. Syst.*, vol. 31, no. 2, pp. 281–305, 2012.

[43] H. Lim, J. Lee, and D.-W. Kim, "Low-rank approximation for multi-label feature selection," *Int. J. Mach. Learn. Comput.*, vol. 6, no. 1, pp. 42–46, 2016.

[44] K. Benabdeslem, H. Elghazel, and M. Hindawi, "Ensemble constrained Laplacian score for efficient and robust semi-supervised feature selection," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 1161–1185, 2016.

[45] P. Yan and Y. Li, "Graph-margin based multi-label feature selection," in *Proc. Eur. Conf. Mach. Learn.*, Riva del Garda, Italy, 2016, pp. 540–555.

[46] V. Kumar and S. Minz, "Multi-view ensemble learning: An optimal feature set partitioning for high-dimensional data classification," *Knowl. Inf. Syst.*, vol. 49, no. 1, pp. 1–59, 2016.

[47] J. Xu, "Effective and efficient multi-label feature selection approaches via modifying Hilbert–Schmidt independence criterion," in *Proc. Int. Conf. Neural Inf. Process.*, Kyoto, Japan, 2016, pp. 385–395.

[48] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, 2009.

[49] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, and N. Sebe, "Web image annotation via subspace-sparsity collaborated feature selection," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1021–1030, Aug. 2012.

[50] J. Wu *et al.*, "Multi-graph-view subgraph mining for graph classification," *Knowl. Inf. Syst.*, vol. 48, no. 1, pp. 29–54, 2016.

[51] L. Jian, J.-D. Li, K. Shu, and H. Liu, "Multi-label informed feature selection," in *Proc. Int. Joint Conf. Artif. Intell.*, New York, NY, USA, 2016, pp. 1627–1633.

[52] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

[53] X. Chang, F. Nie, Y. Yang, and H. Huang, "A convex formulation for semi-supervised multi-label feature selection," in *Proc. AAAI Conf. Artif. Intell.*, Quebec City, QC, Canada, 2014, pp. 1171–1177.

[54] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," *ACM Trans. Knowl. Disc. Data*, vol. 4, no. 3, pp. 1–21, 2010.

[55] H. Li, D. Li, Y. Zhai, S. Wang, and J. Zhang, "A novel attribute reduction approach for multi-label data based on rough set theory," *Info. Sci.*, vols. 367–368, pp. 827–847, Nov. 2016.

[56] J. Xu, J. Liu, J. Yin, and C. Sun, "A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously," *Knowl. Based Syst.*, vol. 98, pp. 172–184, Apr. 2016.
[57] S. Xu *et al.*, "Multi-label learning with label-specific feature reduction," *Knowl. Based Syst.*, vol. 104, pp. 52–61, Jul. 2016.
[58] A. Jalali, S. Sanghavi, C. Ruan, and P. K. Ravikumar, "A dirty model for multi-task learning," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2010, pp. 964–972.
[59] S. Kim, K. A. Sohn, and E. P. Xing, "A multivariate regression approach to association analysis of a quantitative trait network," *Bioinformatics*, vol. 25, no. 12, pp. 204–212, 2009.
[60] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Trans. Cybern.*, to be published.
[61] K. Nag and N. R. Pal, "A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 499–510, Feb. 2016.
[62] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[63] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.
[64] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

Jun Huang received the M.S. degree in computer science from the Anhui University of Technology, Ma'anshan, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China.

He was a Lecturer with the Anhui University of Technology. His current research interests include machine learning and data mining.

Guorong Li (M'15) received the B.S. degree in computer science from the Renmin University of China, Beijing, China, in 2006, and the Ph.D. degree in computer science from the Graduate University of the Chinese Academy of Sciences, Beijing, in 2012.

She is currently an Associate Professor with the University of Chinese Academy of Sciences, Beijing. Her current research interests include object tracking, pattern recognition, cross-media analysis, and multilabel learning.

Qingming Huang (SM'08) received the B.S. degree in computer science and Ph.D. degree in computer engineering from the Harbin Institute of Technology, Harbin, China, in 1988 and 1994, respectively.

He is currently a Professor and the Deputy Dean of the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China. He has published over 300 academic papers in international journals, such as the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and at top level international conferences including ACM Multimedia, International Conference on Computer Vision, Conference on Computer Vision and Pattern Recognition, International Conference on Very Large Data Bases, and International Joint Conference on Artificial Intelligence. His current research interests include multimedia computing, image/video processing, pattern recognition, and computer vision.

Xindong Wu (F'11) received the bachelor's and master's degrees in computer science from the Hefei University of Technology, Hefei, China, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K.

He is a Professor of Computer Science with the University of Louisiana at Lafayette, Lafayette, LA, USA, and a Yangtze River Scholar with the School of Computer Science and Information Engineering, Hefei University of Technology. His current research interests include data mining, knowledge-based systems, and Web information exploration.

Dr. Wu is the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), the Editor-in-Chief of *Knowledge and Information Systems* (Springer), and a Series Editor of the Springer Book Series on Advanced Information and Knowledge Processing. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2005 to 2008. He served as the Program Committee Chair/Co-Chair for ICDM'03, the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2007, and the 19th ACM Conference on Information and Knowledge Management 2010. He is fellow of the AAAS.